



Personal Computing Division

Conexant Access Runner ADSL Linux Drivers

*For the “Tigris” and “Yukon”
Reference Designs*

Installation, Configuration, and Operation Manual

Conexant Access Runner ADSL Linux	
Document Revision:	007.003
Document Level:	Preliminary
Date:	September 10, 2001
Status:	Preliminary
Author:	Jim Webster
File Name:	ConexantLinuxADSL.doc



TABLE OF CONTENTS

1. INTRODUCTION.....	4
1.1 Minimum System Requirements	4
1.2 Compliance	4
1.3 Installing the ADSL Card	4
2. KERNEL UPGRADES	5
2.1 RedHat 6.2 Upgrade	5
2.1.1 Copy the Conexant Distributed Files.....	5
2.1.2 Install the Kernel Source files.....	6
2.1.3 Makefile Setup.....	6
2.1.4 Edit and Run the “lilo” Bootloader.....	6
2.1.5 Install the New Kernel and Support Applications	8
2.2 RedHat 7.1 Upgrade	8
2.2.1 Install the linuxconf Tool.....	9
2.2.2 Copy the Conexant Distributed Files.....	9
2.2.3 Install the Kernel Source files.....	10
2.2.4 Makefile Setup.....	11
2.2.5 Edit and Run the “lilo” Bootloader.....	11
2.2.6 Install the New Kernel and Support Applications	13
3. INSTALLING PPP	14
3.1 PPPoA	14
4. DRIVER INSTALL/UNINSTALL	16
4.1 Driver Installation.....	16
4.2 Driver Uninstallation	16
5. DRIVER CONFIGURATION	17
5.1 Setting the ATM VPI and VCI.....	17



5.2	Configuring the Adaptor.....	17
5.2.1	For Manual (static) IP Configuration:.....	18
5.2.2	DHCP (Dyamnnc IP address) Configuration:	18
5.3	ADSL Line Configuration.....	19
5.3.1	View ADSL Line Configuration.....	21
6.	DRIVER OPERATION	21
6.1	Basic ADSL Driver Operation.....	21
6.2	RFC 1483/2684 LAN Operation	21
6.3	RFC 2512 PPPoE Operation.....	22
6.4	RFC 2364 PPPoA (WAN) Operation	22
6.4.1	Linux Kernel 2.2.16.....	23
6.5	Testing the Data Connection.....	23
7.	RECONFIGURING LINUX FOR YOUR SYSTEM	24
7.1	PPPoATM Configuration.....	24
7.2	Building the Kernel.....	25



1. INTRODUCTION

This Installation, Configuration, and Operations guide is intended for the “Tigris” and “Yukon” ADSL PCI Reference Designs for the Linux operating system on PC-based platforms. The guide instructs you in installing Linux drivers and necessary kernel upgrades, configuring the drivers and protocols, and operating the system.

These instructions assume that: you are somewhat familiar with Linux and a Linux-based editor (such as **pico**); however, detailed Linux expertise is not required. These instructions also assume that “lilo” will be used as the bootloader (which is the default for RedHat installations).

1.1 Minimum System Requirements

The minimum system requirements for ADSL Linux are:

- Pentium 133 Mhz (Pentium II or higher recommended)
- 32 MB RAM (64MB recommended)
- 40 MB free hard disk space
- RedHat Linux 6.2 or 7.1 operating system, with installation disks or files

1.2 Compliance

The ADSL cards and drivers are compliant with following protocols:

- ADSL Physical Layer: compliant with T1.413-1998 or G.992.1 (G.dmt)
- ADSL Encapsulation protocols: RFC 1483/2684 (LAN bridging); RFC 2364 (PPPoA: PPP over ATM) or RFC 2512 (PPPoE: PPP over Ethernet)
- Data Service: IP data over ATM, UBR with AAL5 framing

1.3 Installing the ADSL Card

Before beginning the driver installation, you should remove all power from the PC and install the AccessRunner card, following the PC manufacturer’s instructions for installing new PCI cards. After the card is firmly seated in the PCI slot and the PC has been reassembled, turn power on the PC and verify that the kernel boots up properly. Plug the ADSL line into the “line” jack on the back of the card. Then, follow the instructions below to load the ADSL driver and establish your ADSL connection.



2. KERNEL UPGRADES

To use the Conexant ADSL drivers, you must upgrade your Linux kernel to add support for ATM and the required encapsulation protocols. The drivers will not work on Linux kernels without this support. Currently, we support the Red Hat 6.2 and 7.1 distributions. For RedHat 6.2, the resulting upgraded kernel is 2.2.16. For RedHat 7.1, RedHat recommends upgrading to kernel 2.4.3-12, which we support.

The kernel upgrades are provided as kernel source files that must be compiled using the instructions below. Also available in the Conexant Linux ADSL driver distribution are patch files against specific versions of the 2.2.16 and 2.4.3 kernels, which can be used if you want to install the patches yourself against your own kernel source files. However, installing the patches is recommended only for advanced Linux users and is beyond the scope of this document.

To upgrade from RedHat 6.2, begin with Section 2.1. To upgrade from RedHat 7.1, begin with Section 2.2.

WARNING: The following procedures modify the Linux operating system (O/S) of the target PC. Please back up any valuable data on the target PC and create a boot diskette before proceeding. Special steps must be taken when installing Linux to allow any existing O/S to be retained (for a "dual boot" capability); those procedures are beyond the scope of this document. Also, if these instructions below are not precisely followed, the PC may be left unbootable; then, the O/S must be reinstalled, possibly resulting in a loss of data.

NOTE: To perform kernel upgrades and driver installation, you must always be logged in as the root user.

2.1 RedHat 6.2 Upgrade

We recommend that for best results, you start with a clean installation of RedHat 6.2, following the manual provided, for a "workstation" install. After the install is completed, reboot and log in as root.

2.1.1 Copy the Conexant Distributed Files

Get the following Conexant distributed files (from CD, website, etc.) from the ___ directory and place them in the /usr/src directory of the target PC:

```
atm-0.78-1.i386.rpm
```

```
kernel-atm-headers-2.2.16-5.i386.rpm
```

```
kernel-atm-ppp-2.2.16.tar.gz
```



pppd-2.4.0.tar.gz

2.1.2 Install the Kernel Source files

- Change to /usr/src directory.

```
cd /usr/src
```

- Remove the symbolic link named linux (if it exists).

```
rm -f linux
```

- Install the kernel sources.

```
tar -xzf kernel-atm-ppp-2.2.16.tar.gz
```

- Move the newly created directory “linux” to “linux-2.2.16” and create a symbolic link to it:

```
mv linux linux-2.2.16
```

```
ln -s linux-2.2.16 linux
```

2.1.3 Makefile Setup

Setup the kernel Makefile environment for your system. To accomplish this, run the kernel configuration program “menuconfig”, perform a Save, then perform an Exit. DO NOT change anything.

```
make menuconfig (or make xconfig from an X11 window)
```

2.1.4 Edit and Run the “lilo” Bootloader

Using a Linux text editor such as “pico”, edit the file “/etc/lilo.conf” as follows, and then run lilo to install the new bootload.

Change default boot kernel and add new kernel image entry. The following is an example of a “lilo.conf” file that indicates the lines to add or change. Your file may be slightly different; however, you should make the same changes.

**Notes for editing:**

- Lines to be changed begin with a '>'
- Lines to be added begin with a '+'.
• Do not add the '>' or '+' signs to the file.
- Text after the '#' represents comments (not necessary to add)
- "hda\$" is the root partition; use the same one as the old kernel image (ie. "/dev/hda1" (for IDE hard drives) or "/dev/sdc1" (for SCSI hard drives)).
- After editing, save the new `lilo.conf` (overwriting the old copy).

Sample `lilo.conf`:

```
boot=/dev/hda                #Should be your boot device
map=/boot/map install=/boot/boot.b
prompt timeout=50            #timeout before booting default
kernel message=/boot/message
linear
>default=linux-2.2.16        #default boot to new kernel

image=/boot/vmlinuz-2.2.14-5.0 #old kernel image name
    label=linux                #your image may be different
    number
    read-only
    root=/dev/hda$            #dup. this line below
+
+image=/boot/vmlinuz-2.2.16    #new image entry
+ label=linux-2.2.16          #
+ read-only                   #
+ root=/dev/hda$              #use the line from "root=" above
```

- Run `lilo` to set the boot configuration.

```
lilo
```



2.1.5 Install the New Kernel and Support Applications

1. Compile and install new kernel and modules in the linux directory:

```
cd /usr/src/linux
make dep
make clean
make bzImage
make install
make modules
make modules_install
```

2. Change back to the /usr/src directory and install atm and loadable module support apps.

```
cd /usr/src
rpm -Uvh atm-0.78-1.i386.rpm
```

3. Reboot

```
reboot
```

2.2 RedHat 7.1 Upgrade

This set of instructions is for upgrading the RedHat 7.1 distribution kernel. If you are using another Linux distribution, you may skip this section. The kernel upgrade follows RedHat's recommendation for upgrading 7.1 to the Linux 2.4.3-12 Kernel, with encapsulation protocol patches required for ADSL.

We recommend that for best results, you start with a clean installation of RedHat 7.1, following the manual provided, for a "workstation" install. After the install is completed, reboot and log in as root.



2.2.1 Install the linuxconf Tool

This tool is found on the Red Hat CD "Red Hat Linux 7.1 Operating System Binary CD 2 of 2". If installing from the command line, the RPM filename is:

```
linuxconf-1.24r2-10.i386.rpm
```

and is found on the CD in the "RedHat/RPMS/" directory.

If installing from one of the GUI RPM applications, the "linuxconf" tool is in the "Packages/Applications/System" RPM group.

2.2.2 Copy the Conexant Distributed Files

Get the following Conexant distributed files (from CD, website, etc.) from the ___ directory and place them in the /usr/src directory of the target PC:

```
atm-0.78-1-2.i386.rpm  
pppd-2.4.0.tar.gz  
kernel-atm-ppp-2.4.3-12.tar.gz  
devfsd-2.4.3-12.i386.rpm  
kernel-doc-2.4.3-12.i386.rpm  
gcc-2.96-85.i386.rpm  
cpp-2.96-85.i386.rpm  
gcc-c++-2.96-85.i386.rpm  
libstdc++-2.96-85.i386.rpm  
libstdc++-devel-2.96-85.i386.rpm  
gcc-objc-2.96-85.i386.rpm  
gcc-g77-2.96-85.i386.rpm
```



2.2.3 Install the Kernel Source files

- Change to "/usr/src" directory:

```
cd /usr/src
```

- Remove the symbolic link named "linux" (if it exists)

```
rm -f linux
```

- Install the RPMs for the gcc compiler updates (all this can fit on one command line):

```
rpm -Fvh gcc-2.96-85.i386.rpm cpp-2.96-85.i386.rpm gcc-c++-  
2.96-85.i386.rpm libstdc++-2.96-85.i386.rpm libstdc++-devel-  
2.96-85.i386.rpm gcc-objc-2.96-85.i386.rpm gcc-g77-2.96-  
85.i386.rpm
```

- Install the RPMs for the kernel file system daemon updates:

```
rpm -Fvh devfsd-2.4.3-12.i386.rpm
```

- Install the RPM for the kernel documentation:

```
rpm -Uvh kernel-doc-2.4.3-12.i386.rpm
```

- Install the kernel source files.

```
tar -xzf kernel-atm-ppp-2.4.3-12.tar.gz
```

- Rename linux directory to "linux-2.4.3", make a symbolic link, and then change to the directory.

```
mv linux linux-2.4.3
```

```
ln -s linux-2.4.3 linux
```

```
cd linux
```



2.2.4 Makefile Setup

Setup the kernel Makefile environment for your system. To accomplish this, run the kernel configuration program “menuconfig”, perform a Save, then perform an Exit. DO NOT change anything.

```
make menuconfig (or make xconfig from an X11 window)
```

2.2.5 Edit and Run the “lilo” Bootloader

Using a Linux text editor such as “pico”, edit the file “/etc/lilo.conf” as follows, and then run lilo to install the new bootload.

Change default boot kernel and add new kernel image entry. The following is an example of a “lilo.conf” file that indicates the lines to add or change. Your file may be slightly different; however, you should make the same changes.

Notes for editing:

- Lines to be changed begin with a '>'
- Lines to be added begin with a '+'
- Do not add the '>' or '+' signs to the file.
- Text after the '#' represents comments (not necessary to add)
- "*hda\$*" represents the root partition; use the same version as the old kernel image (e.g. "/dev/hda1" (for IDE hard drives) or "/dev/sdc1" (for SCSI hard drives). Do not include the dollar sign.
- After editing, save the new `lilo.conf` (overwriting the old copy).



Sample lilo.conf:

```
boot=/dev/hda                #Should be your boot device
map=/boot/map install=/boot/boot.b
prompt timeout=50           #timeout before booting default
kernel message=/boot/message
linear
>default=linux-2.4.3        #default boot to new kernel

image=/boot/vmlinuz-2.4.2-2  #old kernel image name
  label=linux                #your image may be different
  number
  read-only
  root=/dev/hda$            #dup. this line below
+
+image=/boot/vmlinuz-2.4.3-12 #new kernel image entry
+  label=linux-2.4.3         #
+  read-only                 #
+  root=/dev/hda$           #use the line from "root=" above
```

- Run lilo to set the boot configuration.

```
lilo
```



2.2.6 Install the New Kernel and Support Applications

Compile and install new kernel and modules in the linux directory:

```
make dep
make clean
make bzImage
make install
make modules
make modules_install
```

2. Install the ATM module support apps.

```
cd /usr/src
rpm -U atm-0.78-1-2.i386.rpm
```

3. Reboot

```
reboot
```



3. INSTALLING PPP

A PPP Daemon is needed if you plan to use PPPoA or PPPoE encapsulation modes. Even if you do not plan to use these modes now, you may want to add it for future use. Follow the following steps:

3.1 PPPoA

- Extract the PPP source tree to directory `"/usr/src/ppp-2.4.0b2"`:

```
cd /usr/src  
tar -xzf pppd-2.4.0.tar.gz
```
- From the `"/usr/src/"` directory enter the following commands to compile and install the `pppd` module:

```
cd ppp-2.4.0b2  
./configure  
make  
make install
```
- Check if the device file `"/dev/ppp"` already exists with command:

```
ls -l /dev/ppp
```
- If it is not there, use the following command to create it:

```
mknod /dev/ppp c 108 0
```

This creates a character (unbuffered) special file capable of sending and receiving data.



- Using an editor such as **pico**, edit the file called **/etc/ppp/options**. We suggest using the following settings:

```
user "<username>"
nodefaultroute
name ADSL
nobsdcomp

nodeflate
nopcomp
novj
novjccomp
noipdefault
noaccomp -am
refuse-chap      #(for compatibility with Microsoft servers)
```

This file is used by **pppd** to control **pppoa** behavior (see **man pppd** for more details).



4. DRIVER INSTALL/UNINSTALL

4.1 Driver Installation

- Get the distributed driver RPM file for your product (Tigris or Yukon) and Linux kernel version (2.2.16 or 2.4.3) and place it the "/usr/src" directory:

```
CnxADSL-uuuATM_km.n.r_v.w.x.y-z.i386.rpm
```

where:

m.n.r = Linux kernel version, e.g. 2.2.16, 2.4.3, etc.

uuu = product line: "Tgr" for Tigris, "Ykn" for Yukon

v.w.x.yyy-z = full product version number, where:

v = product line number (6 for Tigris, 18 for Yukon)

w = PCI H/W version number (1 for Tigris, 2 for Yukon)

x = S/W version number (major)

y = build number (minor)

z = RPM release number (for this build))

- Change to /usr/src directory and install the driver RPM:

```
cd /usr/src
```

```
rpm -Uvh CnxADSL-uuuATM_km.n.r_v.w.x.y-z.i386.rpm
```

...where uuu, v, w, x, y, and z are the product and version names/numbers in the RPM filename as described above.

4.2 Driver Uninstallation

To uninstall the driver:

```
cd /usr/src
```

```
rpm -e CnxADSL-uuuATM_km.n.r_v.w.x.y-z
```



where m.n.r is the Linux kernel version, 2.2.16, 2.4.3, etc., and where uuu, v, w, x, y, and z are the product and version names/numbers, as described in section 4.1 above.

5. DRIVER CONFIGURATION

5.1 Setting the ATM VPI and VCI

To set the driver ATM VPI and VCI configuration parameters, use the "cnxadslconfig" command line program. This program modifies the driver configuration parameter settings in the file "/etc/Conexant/cnxadsl.conf".

The following command example sets the driver's VPI to 0 and VCI to 35 (the default settings):
cnxadslconfig --vpi=0 --vci=35

5.2 Configuring the Adaptor

- Run linuxconf to configure new adaptor and configure adaptor 2 in "Basic host information".

```
linuxconf
```

A menu will appear with configuration options. Select the configuration menu for Network adaptor 2 by following the path below:

Config

|-> Networking

|-> Client tasks

|-> Basic host information

(Scroll down to adaptor 2)

Configure interface as either Manual (for static or fixed IP address) or DHCP (for dynamic IP address) using one of the two sample configurations below. (Check with your ADSL service provider if you are unsure which configuration to use.)



5.2.1 For Manual (static) IP Configuration:

Your service provider should supply you with the Domain name, aliases (optional), IP address, and netmask in linuxconf as shown below:

```

*****
|                               | -----Adaptor 2----- |
|                               | [X] Enabled             |
|Config mode                    | (o) Manual () Dhcp  () Bootp |
|Primary name + domain         |                               |
|Aliases(opt)                  |                               |
|IP address                     | XXX.XXX.XXX.XXX         |
|Netmask(opt)                  | YYY.YYY.YYY.YYY       |
|Net device                     | nas1                    |
|Kernel module                 |                               |
|I/O Port (opt)                |                               |
|Irq (opt)                     |                               |
*****

```

5.2.2 DHCP (Dyammnic IP address) Configuration:

Your service provider should provide you with the Domain name. Enter this in linuxconf as shown below:

```

*****
|                               | -----Adaptor 2----- |
|                               | [X] Enabled             |
|Config mode                    | () Manual (o) Dhcp  () Bootp |
|Primary name + domain         |                               |
|Aliases(opt)                  |                               |
|IP address                     |                               |
|Netmask(opt)                  |                               |
|Net device                     | nas1                    |
|Kernel module                 |                               |
|I/O Port (opt)                |                               |
|Irq (opt)                     |                               |
*****

```

- Reboot the computer to start with new kernel, and log in as root.

reboot

The ADSL card will be started as interface "nas1". The host computer may delay or pause at the prompts "Starting cnxadsl" or "Bringing up interface nas1" for about 30 seconds; this is normal.



5.3 ADSL Line Configuration

To configure the ADSL line, use the `cnxadslconfig` program. Note that values will not take effect until the driver is restarted. (e.g. `cnxadslctl.sh stop` and `cnxadslctl.sh start`)

`cnxadslconfig`

This program modifies the settings in the file: `"/etc/Conexant/cnxadsl.conf"`.

Usage:

```
cnxadslconfig --vpi=0 --vci=35 -q0 -w0 -m0 -p100000 -c -h0 -e0  
-aenable -u10 -flog.txt -s100000
```

Cnxadslconfig Parameter definitions:

Query Parameters:

-q Query

0 - query current common settings

1 - query current settings (all)

h - query ADSL headend choices from ANSI T1.413 Annex D

Set ATM Parameters

--vpi VPI – sets the ATM VPI value

--vci VCI – sets the ATM VCI value

Set AutoSense Parameters:

-w Wiring Selection:

0 - automatic wiring detection

1 - selects combo1 wiring

2 - selects combo2 wiring

3 - selects combo3 wiring

4 - selects combo4 wiring



-m Modulation:

- 0 - selects Auto modulation (G.HS preferred)
- 1 - selects Auto modulation (ANSI T1.413 preferred)
- 2 - selects G.HS modulation
- 3 - selects ANSI T1.413 modulation

Set Data Traffic Parameters:

- p** Peak Cell Rate in cells per second (0 for max)
- c** Clears transfer statistics

Set ADSL Headend Parameters:

- h** ADSL headend (vendor) code from ANSI T1.413 annex D
 - 0 - Non-specific
 - 2-51, B6DB - hex value selects specific vendor
(to see vendor codes, type `cnxadslconfig -Qh`)
- e** ADSL headend environment
 - 0 - selects non-specific
 - 1 - selects no line-driver

Set Autolog Parameters:

- a** enables or disables autologging:
 - enable - enable logging
 - disable - disable logging
- f** sets autolog filename
- u** sets autolog update frequency in logs per 10 seconds
- s** sets autolog max file size



5.3.1 View ADSL Line Configuration

To view the driver's current ADSL line configuration settings, use the following command:

```
cnxadslconfig -q0
```

6. DRIVER OPERATION

6.1 Basic ADSL Driver Operation

The following commands allow you to perform start, stop, and status operation of the ADSL drivers.

- To check whether or not the ADSL modem is in Showtime (physical link connected).

```
cnxadslstatus
```

- To check the status of the driver and get basic statistics:

```
cnxadslctl.sh stat
```

- To stop the driver and servers:

```
cnxadslctl.sh stop
```

- To start or restart the driver and servers:

```
cnxadslctl.sh start
```

6.2 RFC 1483/2684 LAN Operation

This is the default mode of operation, if you do not start PPPoE or PPPoA as described below.



6.3 RFC 2512 PPPoE Operation

The following instructions describe a procedure for installing the Roaring Penguin PPPoE Client from the Linux command line. The Roaring Penguin Binary RPM for Red Hat 6.2 (Intel) is available at <http://www.roaringpenguin.com/pppoe/rp-pppoe-3.0-1.i386.rpm>.

Log in as root and enter:

```
adsl-setup
```

1. Enter your PPPoE user name (default is adsl).
2. Enter the Ethernet interface connected to the ADSL modem, e.g.:

```
nas1
```
4. Decide if you want the link to come up on demand, or stay up continuously.
5. Enter the IP address of your ISP's primary DNS server.
6. Enter your PPPoE network password.
7. Choose the firewall rules to use:

```
STANDALONE
```
8. Accept these settings and adjust configuration files.
9. You are now ready to enter the basic driver start, stop, and status commands shown in Section 6.1.

6.4 RFC 2364 PPPoA (WAN) Operation

- Use the following commands to start up the PPPoATM ADSL connection (refer to section 6.4.1 if using Kernel version 2.2.16):

```
cd /home/users/conexant/ppp-2.4.0b2/
```

```
pppd plugin plugins/pppoatm.so <vpi>.<vci> password "<password>"
```



...where “<vpi> and <vci> are the ATM VPI and VCI used by your network, and *password* is your network connect password.

Once the PPPoATM link is up, the `ifconfig` command should show the new **ppp0** interface with the ip address obtained from server.

- To shut down the PPP link use:

```
/home/users/conexant/ppp-2.4.0b2/scripts/ppp-off
```

6.4.1 Linux Kernel 2.2.16

For Linux Kernel version 2.2.16 the PPPoATM support module must be loaded before attempting a WAN connection. From the command prompt type:

```
modprobe pppoatm
```

6.5 Testing the Data Connection

To test the data connection after starting the driver with the correct encapsulation protocol (as described above), you can test the data connection by pinging an existing IP address using the ADSL device. Always add the "-n" option to the ping command:

```
ping -n XX.XX.XX.XX
```

...where XX.XX.XX.XX is a known IP address available on your network to be pinged.



7. RECONFIGURING LINUX FOR YOUR SYSTEM

Use this procedure to reconfigure Linux if you have special hardware that requires a special configuration, other than the Conexant default configuration.

- Run the configuration utility to configure Linux for your computer.
make xconfig .
- Make sure that you set the following features:
 - **Code maturity level options**
 - y Prompt for development and/or incomplete code/drivers
 - **Loadable module support**
 - n Enable loadable module support
 - n Set version information on all symbols
 - y Kernel module loader
 - **Processor type and features**
 - (Select your processor type, e.g. Pentium II)
 - **Networking Options**
 - y Asynchronous Transfer Mode (ATM) (Experimental)
 - y Classical IP over ATM
 - n Do not load ICMP if no neighbour
 - m LAN Emulation (LANE) support
 - m BR2684
 - **File Systems**
 - y DOS FAT fs support
 - y MSDOS fs support
 - n VFAT (Windows 95) fs support

7.1 PPPoATM Configuration

Differences between the 2.2.16 and 2.4.3-12 kernels require that PPPoATM be setup in an OS specific manner. Use the following settings for verison 2.4.3-12:



- **Network device support**
 - y PPP(point-to-point protocol) support
 - y *PPP over ATM (EXPERIMENTAL)*
- PPP**
- y Async Serial Ports/TTY

For kernel version 2.2.16 setup PPPoATM as follows:

- **Network device support**
 - y PPP(point-to-point protocol) support
 - m *PPP over ATM (EXPERIMENTAL)*
- PPP**
- y Async Serial Ports/TTY

7.2 Building the Kernel

NOTE: Turn on or off other features within xconfig as necessary, based on your computer system, software, and installed peripherals.

- Select **Save and Exit** from **xconfig** main menu.
- Change to the **/usr/src/linux** directory and enter the following make commands to build the kernel:

```
make clean
make dep
make bzImage
make install
make modules
make modules_install
```